NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

A GENERAL UTILITY CORRELATION PROGRAM FOR IBM 709 WITH PROVISIONS FOR MISSING DATA

by

Richard C. Sorenson

and

Thomas D. F. Langen

Project Staff

Thelma Brennan
Martha Menz
Helen Ranck

This Study was Supported in Part by

Office of Naval Research Contract Nonr-477(33)

and

Public Health Research Grant M-743(C7)

Principal Investigator: Paul Horst

January 1963

University of Washington
Seattle, Washington

A GENERAL UTILITY CORRELATION PROGRAM FOR IBM 709 WITH PROVISIONS FOR MISSING DATA

The purpose of the program reported in this paper is to compute the basic statistics and symmetrical correlation matrix for variables for which some data may be missing. Inasmuch as the Bureau of Testing is frequently asked for a correlation program for a battery of variables of diffeing score form, provision has been made for varying the format of the data cards. Oftentimes those requesting the program vary from computer sophisticates to statistical novices. This paper is designed to serve the latter and will thus include detail that is not usually found in papers of this nature.

The program will process any number of batches of data with each batch accompanied by its format control and processing control cards. Within each batch of data there may be up to 99,999 cases. The program is limited to one punched card per case and thus the number of variables is limited by the sum of the maximum number of digits required by the scores for all the variables. This sum must not exceed 80. The missing data provision requires two columns so that any variable which has missing data must be written as at least a two digit score even if it be scored originally as one digit. While a case identification or serial number may be punched on the card, this is not necessary to the program and will reduce the columns available for scores.

Format Control Card

A format statement which describes the cards and identifies the variables on which statistics are to be computed is required. This will be described now since it helps one to understand the card form. This statement consists of a collection of symbols in the two forms NX and NFA.B. The statement begins and ends with parentheses (the first parenthesis is placed in column 1) and each symbol is separated from the following one by a comma. The NX symbol designates successive columns which are to be ignored in the computations, "N" designating the number of successive columns concerned and "X" directing that they be ignored. The NFA.B symbol designates successive columns for which computations are desired and describes the score form involved, "N" designating the number of successive similar score forms, "F" directing that computations be done, "A" designating the number of columns in the score, and "B" designating the number of columns to the right of the assumed decimal point. The format statement must account for all columns of the card up to and including the last one containing data to be correlated.

A. Illustration: A ten digit serial number and seven successive scores of the form xxxx.xxxxxx (with the decimal point not punched).

Example: (10X,7F10.6)

B. Illustration: No serial number, six successive scores of the form xx.xx (with the decimal point not punched) twenty blank columns, seven successive scores of the form xxxx, four successive scores or entries of the form xx which are not to be correlated.

Example: (6F4.2,20X,7F4.0,8X) or (6F4.2,20X,7F4.0)

C. Illustration: A four digit serial number, three successive scores of the for xxx, five blank columns, ten columns containing several variables which are not to be correlated, six scores of the form 0.xxxxx (with "0" and the decimal point not punched) and four successive scores of the form x. (It is assumed that the six successive scores of the form 0.xxxxx were all less than 1.0 and thus the "0" was not required to be punched.)

Example: (4X,3F3.0,15X,6F5.5,4F1.0)

D. Illustration: Same as C except that "0" and decimal point were punched.

Example: (4X,3F3.0,15X,6F7.5,4F1.0)

In the illustrations and examples shown above, it will be noted that no provision has been made for signs. This program will not accept negative scores because of the provisions for missing data. As mentioned this provision requires two columns, the first to be punched with a minus sign and the second with a "1". The program would treat any negative score as missing data rather than as a score. Where there are negative scores, one should add a constant to all scores in that variable to bring the lowest score to a positive value. Correction of the various outputs for the presence of this constant is relatively simple and will be outlined later. In Illustrations C and D above, no provision was made for missing data in the final four one-digit scores. If there were missing data in these four scores, it would have been necessary to allow eight columns instead of the four provided. This could have been done by utilizing the empty columns at the right of the card and recording the one-digit scores as two-digit scores in the form x0. The example of the format statement would then be (4X,3F3.0,15X,6F5.5,4F2.1) for C and (4X,3F3.0,15X, 6F7.5,4F2.1) for D. Where the score form consists of three or more digits, the minus must be punched in the first column of the data field followed by a "1" with the remainder of the field punched "0".

Processing Control Card

The foregoing discussion should give one a good view of the flexibility of the program in regard to the format of the input data cards and the format control card. Now let us consider the processing control card. This card allows the user to communicate to the program some very important information. In columns 1-2 one must punch the number of variables for which the program is to calculate statistics. This figure may take on values from 1 to 80 inasmuch as there are 80 columns on a card. Of course one would normally wish to have more than one variable. In columns 3-7 one must punch the number of cards (number of cases or entities). This number must be less than 100,000.

A. Illustration: Data for 965 cases on 32 variables

Example: 3200965

B. Illustration: Data for 2023 cases on 25 variables

Example: 2502023

Order of Card Input

When the cards are submitted to a computer center for processing the order of the cards is of particular import. The cards must be in the following order:

1) Cards required by the computer center
2) Symbolic or binary program card deck
3) * DATA card (required by computer center for monitor runs)
4) Format control card
5) Processing control card
6) Data cards (order within data cards is not significant)
7) Format control card (for second batch of data if any)
8) Processing control card (for second batch of data if any)
9) Data cards

There may be as many batches of data as are needed. The last batch of data cards must be followed by 2 blank cards. As an illustration let us suppose we had three different studies which involved correlating a number of variables.

The data for the first study included scores on eight tests for 1232 students. The data is to be punched allowing three columns for each score. The second study concerns ratings of 75 students by professors with regard to 3 characteristics. The data is in the form of the average rating (to the nearest value in second decimal place) with two digits to the left of the decimal point, and two digits to the right of it (the decimal point is punched). A ten digit case identification is punched in the first ten columns. For the third study we have preference data from a survey of people picked at random. We have data for 89,786 people and the data is complete on ten questions but 101 cases are missing on questions eleven and twelve. The data is coded "0" for "dislike", "1" for "indifferent", and "2" for "prefer".

If these data were to be submitted at the University of Washington Research Computer Laboratory the following cards would be needed:[1]

1. ********* RUN SEQUENCE NUMBER ***** 000000 *********************
2. *NNNNNNN LAST NAME, FIRST NAME
3. * XEQ
4. * MAX TIME (MMM)
5. Program card deck
6. * DATA
7. (8F3.0)
8. 0801232
9. Data cards for the first batch

[1] It is recommended that one consult the job submitting manual of the computer center as these required cards are often modified. The run sequence number card, *XEQ,*MAX TIME, and * DATA cards are obtained from the computer center.

10. (10X,3F5.2)

11. 0300075

12. Data cards for the second batch

13. (10F1.0,2F2.0)

14. 1289786

15. Data cards for the third batch

16. Blank card

17. Blank card

The NNNNNNNN of the name card is the RCL job number, the MMM of the MAX TIME card is the maximum time in minutes.

Output

Since there are missing data possible on any variable, the program computes its statistical values for each variable as though it were paired with each other variable (which it is) and thus there will be as many means, for example, for each variable as there are variables.

The output will be as follows:

(a) The first column identifies the "i" variable and the second the "j" variable of the "ij" pair of variables.

(b) Next comes r_{ij} expressed in form 0.xxxxxE YY. The "YY" indicates the number of places the decimal point is to be moved; if the value "YY" is preceded by a minus sign, move the decimal to the left; otherwise, move the decimal to the right. Of course the decimal will not be moved if "YY" is "00".

Examples:	0.10000E 01	means	1.0000
	0.18732E-02	means	0.0018732

0.23456E 00 means 0.23456

0.45678E-00 means 0.45678

A minus sign before the 0 in the 0.xxxxxE YY means a negative coefficient.

(c) Sums of cross products for the paired variables. These will be in the form x.xxxxxxE YY, with the same meaning for "E" and "YY" as given above.

(d) Mean of the i variable in the same form and meaning as given for (c). If there were no missing data, all means of "i" would be identical.

(e) Mean of the variable in the same form and meaning as in (d).

(f) Standard deviation of the i variable in the same meaning as above but in the form x.xxxxE YY. If there were no missing data, all standard deviations of "i" would be identical.

(g) Standard deviation of the j variables in the same form and meaning as in (f) above.

(h) The number of cases common to the paired variables.

(i) Sums of the squares of scores for the variable in the form x.xxxxxxxE YY with the usual meaning.

(j) Sums of the squares of scores for the j variable in the same form and meaning as in (i) above.

Items (b) through (j) may be clearer if we consider the problem of forty cases with four variables but with two missing scores on one, three missing scores on the second, one missing score on the third, and none missing on the fourth. For such a problem N_{12} may be 35, 36, or 37; N_{13} may be 37 or 38; N_{14} would be 38; N_{23} would be 36 or 37; N_{24} would be 37; and N_{34} would be 39.

This is probably an appropriate time to discuss the correction of the various outputs for any constant that was added to scores to insure all positive values. The number of cases, the correlation coefficients, and the standard deviations were not affected by the constant. Subtraction of the constant from the printed mean will give the correct mean. The corrected sum of scores is the number of cases times the corrected mean. The sum of squares of scores for a given variable is corrected by subtracting both the square of the constant times the number of cases and the product of twice the constant times the corrected sum of scores. The sum of cross products is corrected by subtracting these products: the constant for the first variable times the corrected sum of scores of the second variable, the constant for the second variable times the corrected sum of scores for the first variable, and the number of cases times the product of the two constants.

Time Estimate

One may gain an approximation of the time needed for execution of a group of problems by the following formula:

Let C_i = the number of cases for the ith batch

V_i = the number of variables in the ith batch

then the total execution time in seconds will be approximated by

$$T = 30 + .751(\sum_i C_i + \sum_i V_i^2)$$

For a run including the first two of the three batches of data in our example the time estimate would be calculated as follows:

$C_1 = 1232, V_1 = 8$

$C_2 = 75, V_2 = 3$

$T = 30+.751(1232+75+8^2+3^2)$

$T = 30+.751(1380) = 1066.4$ seconds

This time estimate is not precise and will give an overestimate for batches where C_i is less than V_i^2. The maximum time to be punched as MMM on the MAX TIME card should exceed the expected time in minutes by 10% to 20%.

Program of Instructions

(FORTRAN Symbolic Form)

```
     DIMENSION FMT (14)
     DIMENSION S(3240,6), D(100,80)
     COMMON S
 006 READ INPUT TAPE 5,002,(FMT(I),I=1,14)
  31 FORMAT (2I3,E13.5,1PE14.6,2E13.6,2E12.4,0PF7.0,1P2E15.7)
  40 FORMAT (2I3,13H NOT DEFINED 1PE14.6,2E13.6,2E12.4,0PF7.0,1P2E15.7)
 002 FORMAT (13A6,1A2)
     READ INPUT TAPE 5,18,N,M
     IF (N) 5,300,5
  18 FORMAT (I2,I5)
 005 NR=(M/100)
     NT=M-(NR*100)
     NQ=100
     NR=NR+1
     IRCS=N*(N+1)/2
     DO 620 ICRS=1,IRCS
     DO 620 ISCR=1,6
 620 S(ICRS,ISCR)=0
     DO 621 ICRS=1,100
     DO 621 ISCR=1,80
 621 D(ICRS,ISCR)=0
     DO 310 KM=1,NR
     IF(KM-NR) 124,125,124
 125 NQ=NT
 124 DO 126 I=1,NQ
     CALL SAVIO
     READ INPUT TAPE 5,FMT,(D(I,J),J=1,N)
 126 CALL NSAVIO
     DO 130 I=1,NQ
     DO 112 J=1,N
     IF(D(I,J)) 112,111,111
 111 DO 109 K=J,N
     IF (D(I,K))   109,108,108
 108 IS=N*J-N+K+(J-J**2)/2
     S(IS,1)=S(IS,1)+D(I,J)*D(I,K)
     S(IS,2)=S(IS,2)+D(I,J)
     S(IS,3)=S(IS,3)+D(I,K)
     S(IS,4)=S(IS,4)+1.
     S(IS,5)=S(IS,5)+D(I,J)**2
     S(IS,6)=S(IS,6)+D(I,K)**2
 109 CONTINUE
 112 CONTINUE
 130 CONTINUE
 310 CONTINUE
1783 FORMAT (120H  I  J     R(I,J)      SUM X(I)*X(J)    MEAN X(I)     MEA
    1N X(J)     SD X(I)      SD X(J)       N    SUM X(I)**2   SUM X(J)**2
```

```
     2)
260 WRITE OUTPUT TAPE 6,1783
    DO 717 J=1,N
    DO 717 K=J,N
    IS=N*J-N+K+(J-J**2)/2
    SDI=SQRTF(S(IS,4)*S(IS,5)-S(IS,2)**2)
    SDJ=SQRTF(S(IS,4)*S(IS,6)-S(IS,3)**2)
    EVI=S(IS,2)/S(IS,4)
    EVJ=S(IS,3)/S(IS,4)
    RD=SDI*SDJ
    DN=SQRTF(S(IS,4)*(S(IS,4)-1.))
    SDI=SDI/DN
    SDJ=SDJ/DN
    IF (RD) 271,270,271
270 WRITE OUTPUT TAPE 6,40,J,K,S(IS,1),EVI,EVJ,SDI,SDJ,S(IS,4)
    GO TO 717
271 R=(S(IS,4)*S(IS,1)-S(IS,2)*S(IS,3))/RD
    WRITE OUTPUT TAPE6,31,J,K,R,S(IS,1),EVI,EVJ,SDI,SDJ,S(IS,4),S(IS,5
   1),S(IS,6)
717 CONTINUE
    GO TO 006
300 CONTINUE
     CALL    EXIT
     END
```

NOTE:

The program directs that a check be made at the time each card is read for obviously mispunched cards, e.g., alphabetic characters in numeric fields. If a card is incorrect it will be listed on the output and that case will not be included with the data in making calculations. Extreme care must be taken in punching the format control card inasmuch as an error in the format may result in the program printing the data out card by card with the explanation that it is not appropriately punched.

Two subroutines, SAVIO and NSAVIO are called and are expected to be on library tape. If they are not on library tape at the computer center to which you have access the CALL SAVIO and CALL NSAVIO cards may be removed from the program without hampering the functioning of the rest of the program. They have to do with the aforementioned printing out of incorrect cards and allow the program to continue execution. Information concerning these subroutines may be had from the Research Computer Laboratory, University of Washington.